REMARKS

Claims 2-13, 15-21 and 23-32 are pending in the application, and have been rejected.

Rejections under 35 U.S.C. § 103 over Hatanaka in view of Ardoin

Claims 2-13, 15-21, 23-28, 30, and 32 stand rejected under 35 U.S.C. § 103(a) over Hatanaka (U.S. Patent No. 5,923,573) in view of Ardoin et al. (U.S. Patent No. 5,692,184). Claims 4, 15, 18, 30, and 32 are the independent claims of this group. Hatanaka discusses a CAD system used with "kit models," which are pre-made three-dimensional wire frame models formed by a group of lines. *See* Hatanaka, Col. 5, lines 28-32. These kit models appear to be generic models to which a designer can add details or make changes. Each of the wire frame lines intersects other lines. *Id.* A designer can input "modification information" to modify details of the wire frame lines (including "characteristic lines," which are lines the designer adds to the model). A modification routine 16 then changes individual characteristic lines. *See id.*, Col. 6, line 58 to Col. 7, line 6. The patent indicates that a "model regeneration function" will cause elements that are related to an object element to be changed when the object element is changed. *See id.*, Col 7, lines 18-19, 28-32, 43-45. The patent also discusses a particular manner in which relationships among elements can be tracked using "generation relationship data." *See id.*, Col. 7, line 56 to Col. 8, line 67. To regenerate a modified model, modification information is received from an operator, and elements associated with the modified elements are determined using the relationship data. *Id.*, Col 9, lines 12-25. Items known as "F nodes" for each element that are affected by a change are put into an "F node queue" 161a, and "generation functions" relating to F Nodes of particular elements are executed until the queue is empty, and the regeneration is complete. *Id.*, Col. 9, lines 26-56; Col. 9, line 57 to Col. 10, line 54 (describing generation functions). Thus, the system identifies elements to be updated, checks the F nodes for those elements into an F node queue, and executes generation functions for each element.

The Office Action equates the generation relationship data of Hatanaka with the "first step" creation of the pending claims. *See* Office Action, at 4. However, the generation relationship data of Hatanaka is merely information regarding the correlation between one

graphic element and other graphic elements (such as "ingredient elements"). As Hatanaka notes: "[T]he generation relationship data 24 retains information as to with which and how the geometric shapes of the graphic elements are generated." Hatanaka, Col. 8, lines 3-5. Therefore, it is not a step.

The office action recognizes that Hatanaka fails to disclose creating a second step, and that it more plainly fails to disclose a second step that is based on the first step and the structure of one of the elements or the relationship between two of the elements. In addition, the office action recognizes that Hatanaka does not disclose the sorting of steps before executing them. Yet the office action concludes that providing a second step based on the first step and the structure of one of the elements or the relationship between two of the elements would have been inherent or obvious. To support the conclusion of obviousness, however, the office action does not point to any motivation in the prior art to modify Hatanaka in such a manner. The office action appears to assume that it is enough for the prior art simply to disclose two steps, regardless of what they are based on. That is not the case, because the steps of the claims are generated according to particular information. Claim 4, for example, recites that the first step is to be based on the structure of a first element or a relationship between the first element and another element or elements, and the second step is to be based on a combination of the first step and the structure of one of the elements or a relationship between two of the elements. The Hatanaka reference at most suggests the use of multiple steps, but does not disclose or suggest what those steps are based on, and certainly does not note that the second step is based in part on the first step. Rather, even granting Hatanaka a liberal reading, it at most might suggest that the steps are all based on the change in an initial element (perhaps as relating to "generation relationship data"), but not that subsequent steps are dependent on previous steps. This relative relationship between steps, in combination with the relationship between later steps and elements, simply is not shown or even suggested in Hatanaka, and the office action does not address the point.

The disclosure of the generation of multiple steps ***based on*** the factors recited in the claims in also not inherent in Hatanaka. Even assuming that Hatanaka generates two steps, there

is no reason that the steps could not be based on some factors other than those recited in the pending claims, so that Hatanaka does not inherently disclose the recited features. Also, Hatanaka does not inherently disclose generation of multiple steps in any form. Specifically, Hatanaka could operate by changing a model each time a user makes an alteration to the model, so that there would be no need to generate multiple steps; each change would be accompanied by a single step. Because Hatanaka could operate with single steps, it does not provide inherent disclosure, and because it does not suggest multiple steps (or multiple steps based on the factors recited in the claims), it is not an appropriate reference for an obviousness rejection.

With respect to sorting of steps before executing the steps, Ardoin simply discusses the ordering of "nodes," which are items that "can represent a value, an associative evaluation function or a predicate." Ardoin, Col. 6, lines 34-37. Ardoin does not indicate either explicitly or by inference that these nodes are "steps," as that term is used in the pending application and recited in the claims. In essence, the office action's position is that sorting of ***any*** entity related to a CAD application is sufficient to teach the recited limitations. But claim 4, for example, recites sorting of steps before executing steps. As the pending application explains, this sorting procedure can aid the efficiency of drawing regeneration in a modeling system by placing steps in the best order for proper execution. *See, e.g.,* application, page 13, lines 17-24; page 18, lines 15-22. Nothing in Ardoin discloses or suggests sorting for such a purpose or even the sorting of steps.

From the limited disclosure available in Ardoin, it appears that nodes are ***not*** steps or anything similar to steps, but are instead simply units of relationship between elements. Thus, nodes simply assist in describing the static condition of a model, but are not steps or even equivalent to steps taken to effect changes in a model.

Furthermore, the office action shows no motivation to combine Hatanaka with Ardoin, or that the combination, if made, would result in the claimed invention. First, Hatanaka does not even disclose multiple steps, so there would be no reason for a skilled artisan to look for any sort of sorting reference—even the unrelated sorting shown in Ardoin. Thus, there is no motivation to combine. Even if the artisan was told to combine Hatanaka with Ardoin, he would at most get

an invention with multiple sorted elements, and a single step, since Ardoin relates at most to the sorting of static elements. The skilled artisan would not end up with multiple elements, multiple sorted steps (which are each based on particular factors not even mentioned in Ardoin), and execution of the steps to change at least one element.

Thus, claim 4 and the claims that depend on it are in proper condition for immediate allowance.

Claim 15 recites a system for regenerating a design model, comprising a model element and a step propagator that receives a first step that represents changes in the model element, and produces a second step that represents other changes in the model element that are dependent on the first step. The system also comprises a step executer that executes the first step and the second step, and a step sorter that sorts the first step and the second step according to dependencies between the steps. As just mentioned, Hatanaka does not disclose multiple steps, Ardoin discloses only sorting of static elements and not the sorting of steps, and the office action has pointed to no motivation to combine the references. Therefore, claim 15 and the claims that depend on it are in condition for allowance, and the applicants request the same.

Claim 18 recites a method of propagating changes through a plurality of elements in a model, comprising analyzing changes in a first element, generating a first step to carry out at least some of the changes in the first element, and generating a second step based on a predefined relationship between the first element and one or more other elements, or on changes in a predefined relationship between the first element and one or more other elements. The method also involves executing the first step and the second step on the plurality of elements to reflect the changes in the first element and the relationship between the first element and the one or more elements, and sorting the first step and the plurality of steps to ensure that each step is executed after steps on which it depends are executed. As explained, neither Hatanaka nor Ardoin disclose or even fairly suggest such a combination of elements, and claim 18 is also in condition for immediate allowance.

Claims 30 and 32 also recite methods involving the sorting and execution of steps in updating data in a model, and are patentable for the reasons just discussed.

The dependent claims differ even more from the prior art, and the office action does not indicate how Hatanaka or Ardoin disclose the features of these claims either. For example, claim 5 recites sorting by a depth-first sorting method. The office action admits that neither reference discloses this feature (see page 4), but asserts that the feature is "equivalent in functionality" to the sorting of Ardoin. First, "equivalence in functionality" is not the test for patentability so this assertion is off the mark from the start. Second, Ardoin does not show any sorting of steps, so it is hard to understand how it would be able to disclose depth-first sorting of steps.

As another example, claim 6 recites storing the steps in a step repository. Again, the office action admits that neither Hatanaka nor Ardoin disclose or even suggest such a feature. Instead, the office action speculates that the "generation of Hatanaka must be stored in some sort of memory," and that such would be "substantially similar in functionality" to what is claimed. *See* Office Action, page 18. Again, the office action has not applied the appropriate legal standard for obviousness. Also, Hatanaka discloses only one step, so there is no reason to believe that it would need to, or even want to, store any steps in a particular area of memory dedicated to storing of steps for execution. Rather, it is more reasonable to conclude that Hatanaka simply executes the step—there is certainly nothing in Hatanaka to suggest otherwise. Even if Hatanaka disclosed multiple steps, it could store them separately, such as with their associated elements. Thus, even granting Hatanaka its broadest possible reading, it does not disclose or fairly suggest the claimed features.

As yet another example, claims 7 and 8 recite, respectively, a nul step and a nul step that instigates regeneration. (Claim 23 has a similar feature.) On this point, the office action simply points to a definition of "null cycle." All this citation does is establish that null cycles exist, which the applicants have never disputed. Rather, as the applicants pointed out in the prior response, there is no suggestion in the prior art (or in the dictionary definition) regarding the particular use of a nul step as recited in the claims. In particular, the dictionary simply indicates that there are times in which a microprocessor is working without receiving new data. That does not disclose in any way the use of an actual positive step in a method that is a nul step, and certainly does not suggest that the nul step should be used as an instigator of regeneration.

Indeed, it seems to suggest the opposite—that the null cycle (which is not a nul step) should have no outside interaction. Also, even if modeling programs have a "refresh" command, there is no reason to extend that simple fact to a conclusion that it would have been obvious to use a nul step as a regeneration instigator. Indeed, the conclusion appears to be based entirely on improper use of hindsight that uses the applicants' disclosure as a template for applying the prior art.

On dependent claims 9 and 17, the office action admits that the applied art does not disclose or suggest the recited feature, but asserts without authority that the invention would have been obvious. This statement, without support, does not make a prima facie case, and applicants respectfully request allowance of claims 9 and 17 also.

For claim 10, Hatanaka does not show multiple steps, so it cannot show a second step that depends on the relationship between a first element and a class of elements.

As yet another example, claim 21 recites a second step that is selected from a plurality of steps based on the generation of other steps. Because Hatanaka shows only one step, and Ardoin at most shows sorting of elements rather than sorting of steps, there is no way that either reference could show this particular way of selecting steps for execution. Indeed, the office action recognizes as much, as it admits that neither reference discloses this feature, and provides only an unsupported conclusion that the feature would have been obvious.

Claim 25, which recites generation of the steps by prediction, provides another example. The office action supposes that the user of the Hatanaka or Ardoin references would have to "predict" how the model is going to look. Such "prediction" is irrelevant, however. It has nothing to do with what the propagation method is doing. It also has nothing to do with steps. There is a critical difference between predicting what the model might look like and generating (and executing) the steps to get it to its ultimate changed point.

Claim 27 also provides another example of an unsupported conclusion in support of a rejection.

For each of these reasons, the applicants respectfully submit that claims 2-13, 15-21, 23-28, 30, and 32 are all in clear condition for allowance. As a result, the applicants request a speedy allowance of the claims.

Rejections under 35 U.S.C. § 103 over Hatanaka in view of Ardoin and Pabon

The Examiner rejected claim 29 under 35 U.S.C. § 103(a) as being unpatentable over Hatanaka (U.S. Patent No. 5,923,573) in view of Ardoin et al. (U.S. Patent No. 5,692,184), and further in view of Pabon (U.S. Patent No. 5,251,290). Claim 29 depends on claim 18, and recites the feature of "verifying the elements after execution for constraint satisfaction."

As an initial matter, claim 29 is patentable because it incorporates the features of claim 18, which are not disclosed or fairly suggested by Hatanaka or Ardoin (as discussed above), or by Pabon. The office action admits that Pabon also does not disclose the feature added to claim 18 by claim 29. *See* Office Action, at 13. Thus, the rejection is improper from the outset because no reference discloses the claim feature. Also, Pabon does not fairly suggest the recited feature simply by indicating that there are "constraint driven problems" in modeling, as because such a bare disclosure does not have anything to do with model regeneration, and the office action does not point to any motivation to combine Pabon with the other references. Rather, the rejection is simply a use of the applicants' disclosure to search prior art databases for the term "constraint," and then to use applicants' disclosure as a template for a hindsight reconstruction of the invention using totally unrelated pieces of prior art.

Rejections under 35 U.S.C. § 103 over Hatanaka in view of Hollingsworth

The Examiner rejected claim 31 under 35 U.S.C. § 103(a) as being unpatentable over Hatanaka (U.S. Patent No. 5,923,573) in view of Hollingsworth (U.S. Patent No. 5,444,836).

Claim 31 is an independent claim that recites a method of propagating changes made in one data element to a related data element, comprising accumulating changes made to the one data element, identifying a predetermined number of possible mutually-exclusive sets of changes that may be made to a related data element, selecting the most appropriate set of changes by employing a predetermined selection standard, and testing the selected set of changes to determine whether it is an appropriate set of changes. The Examiner asserts a belief that Hollingsworth accumulates changes in an element, but does not point to anything in

Hollingsworth that teaches or suggests such accumulation. Rather, Hollingsworth simply discloses a system that uses certain rules for the placement of objects. Also, Hollingsworth's discussion of "overplotting" does not, as the Office Action maintains, indicate that any sort of selection of one set of changes is being made from among a group of possible changes. Rather, it merely shows the failure of a rule. Hollingsworth does not "select" the next rule after a failure; rather, it is forced to use the rule. This is known as a Hobson's Choice, which is no choice at all. It is not a selection. In any event, the office action does not indicate anywhere in the prior art the accumulation of changes and the identification of mutually-exclusive sets of changes. Thus, claim 31 recites allowable subject matter for multiple independent reasons.

Please charge deposit account 06-1050 in the amount of $55.00 for the Petition for Extension of Time fee. Please apply any other charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: 6-25-04

John A. Dragseth
Reg. No. 42,497

Fish & Richardson P.C., P.A.
60 South Sixth Street
Suite 3300
Minneapolis, MN 55402
Telephone: (612) 335-5070
Facsimile: (612) 288-9696

60202096.doc